

Software-Management-Systeme erfreuen sich einer immer größeren Beliebtheit. In diesem Artikel werden die Gründe für den Einsatz eines Software-Management-Systems zur Verwaltung der Clients in der zentralen Verwaltung der Humboldt-Universität dargelegt. Anhand des Softwarepaketes OpenOffice wird das System „Baramundi-Management-Suite“ der Firma Baramundi vorgestellt.

Ausgangssituation

Die Verwaltung der Client-Systeme im Verwaltungsnetz der Humboldt-Universität gehört zu den aufwendigsten Aufgaben, welche im Rahmen der Systemverwaltung anfallen. Dies ist in der Tatsache begründet, dass die Aufgaben der Verwaltung nur in geringem Maße eine Beschränkung auf einheitliche Software zulassen. Bisher wird die Bereitstellung von Rechnern über drei aufeinander folgende Prozesse abgebildet. Bei diesen Prozessen handelt es sich um die Installation des Betriebssystems, die Personalisierung des Rechners und ein Update-Verfahren für Betriebssystem und Software. Nachfolgend werden die Prozesse genauer beschrieben.

Erster Prozessschritt Remote Installation Service

Die Installation des Betriebssystems erfolgt über die Remote Installation Services (RIS) von Microsoft, einem optionalen Bestandteil von Microsoft Windows Server, der ab der Serverversion 2000 verfügbar ist. Mit Hilfe dieser Serverkomponente werden das Betriebssystem, die Computerkonfiguration und die Anwendungsprogramme bei einer Erstinstallation des Rechners über das Netzwerk auf dem Rechner installiert. Vor dem Start der Installation, erfolgt eine manuelle Partitionierung der Festplatte, da die RIS-Server-Installation keine Partitionierung zulässt. Für die Installation wird ein benutzerdefiniertes Image mit allen Konfigurationen und benötigten Treibern erzeugt und im RIS-Server implementiert. Mit Hilfe einer Antwortdatei, in der Informationen wie z. B. der Lizenzschlüssel

hinterlegt sind, läuft die Installation ohne Nutzereingriffe ab. Die Verarbeitung erfolgt sequenziell, so dass im Anschluss an die Betriebssysteminstallation die automatisierte Installation der Standardprogramme der Verwaltung stattfindet. Bei den in der Installations-Routine definierten Konfigurationen handelt es sich unter anderem um Netzwerkeinstellungen, Systemrechte, Benutzerkonten, Desktop-Verknüpfungen, Firewall-Einstellungen sowie die Installation von Programmen, mit denen jeder Rechner ausgestattet wird. Als Beispiele seien MS-Office 2003, Adobe Acrobat Reader oder Mozilla Firefox und Thunderbird genannt. Weitere Programminstallation, die über den Standard hinausgehen, erfolgen über den Update-Service. Dieses, in einem Projekt selbstentwickelte Verfahren nutzt einen über den RIS-Server installierten speziellen Dienst.

Zweiter Prozessschritt Personalisierung

Zum Personalisierungsprozess gehören neben der Vergabe von festen IP-Adressen die Aufnahme der Rechner in die Windows-2003-Domäne und die Anpassung an die benutzerspezifischen Anforderungen des Arbeitsplatzes. Besonders zeit- und arbeitsaufwendig sind die Installationen der zusätzlichen nutzerspezifischen Software und Peripheriegeräte vor Ort.

Dritter Prozessschritt Update-Service

Als letzter Prozessschritt findet das Softwareupdate-Verfahren statt. Bei diesem

Verfahren handelt es sich um eine Eigenentwicklung der Arbeitsgruppe zur Softwareverteilung. Es basiert auf einer Vielzahl von Skripten, die über mehrere Jahre hinweg entstanden sind.

Der über den RIS-Server auf dem Client installierte Update-Dienst startet automatisch bei jedem Login am Rechner und verbindet sich über das Netzwerk mit dem Update-Server.

Die entwickelten Algorithmen stoßen bei jedem Start des Rechners eine umfangreiche Prüfroutine an, welche die vorhandenen Skripte und Programme aktualisiert. Ein Teil der Programminstallation erfolgt IP-basiert auf den Rechnern. Mittels eines Systems von Log-Dateien werden zeitliche Abläufe gesteuert und die Ausführung der Skripte überwacht. Die nachfolgende Grafik veranschaulicht den Prozess:

führen. Da gleichzeitig ein Verfahren zur Inventarisierung von Hard- und Software benötigt wurde, rückten Client-Management-Systeme in den Fokus der Arbeitsgruppe Systembetreuung/Windows. Bei Client-Management-Systemen handelt es sich um Anwendungen, die eine effiziente Verwaltung und Inventarisierung von Hard- und Software in IT-Umgebungen ermöglichen.

Nach einer eingehenden Marktrecherche entschied sich die Arbeitsgruppe Systembetreuung/Windows der Abteilung „DV in der Verwaltung“ für das System der Firma Baramundi. Dieses Produkt erfüllt die gestellten technischen Anforderungen wie Skalierbarkeit der Komponenten, die Abbildung vorhandener Strukturen in das System und ein sehr differenziertes Rechtemanagement, um Arbeitsaufgaben delegieren zu kön-

nen. Weitere Entscheidungskriterien waren die Offenlegung der Datenbank, die weitgehende Anpassungen ermöglicht, sowie die Unternehmensgröße, die es erlaubt, auch als „kleiner“ Kunde Änderungswünsche zu platzieren. Der Fokus bei der Auswahl des Systems lag auf der Systembetreuung, für die eine technische und organisatorische Strukturreform angestrebt wurde.

Baramundi

Bei Baramundi handelt es sich um ein klassisches Client-Management-System, das mit Reporting- und Client-Recovery-Funktionen gekoppelt ist. Es fällt durch seine Vielzahl von integrierbaren Funktionen auf. Die „baramundi-Management-Suite“ verfügt über eine benutzerfreundliche, in die Microsoft-Management-Konsole integrierte Administrationsoberfläche, die eine weitgehende Delegation von Funktionen und Rollen ermöglicht.

Systemarchitektur

Wie schon im Artikel „Inventarisierung“ beschrieben, handelt es sich bei der „baramundi-Management-Suite“ um eine Modul-Lösung. Die Module sind einzeln einsetzbar und über eine gemeinsame Administrationsoberfläche bedienbar. Die Lösung ist hoch skalierbar und umfasst alle Komponenten, die

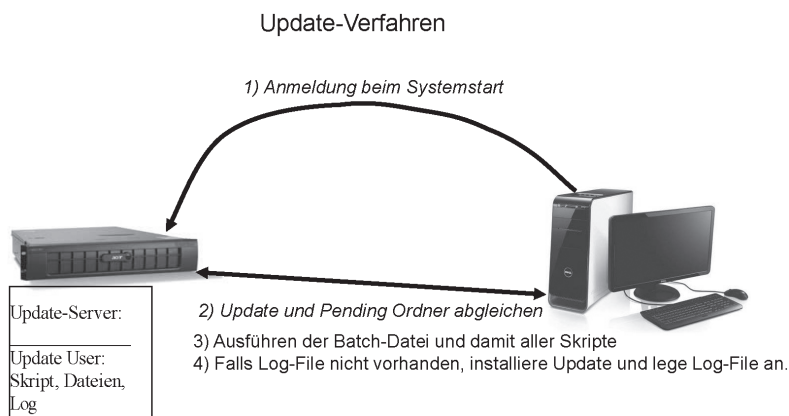


Abb. 1: Update-Verfahren

Je nach Komplexität der Algorithmen kann das hier beschriebene Verfahren längere Zeit in Anspruch nehmen. So können unter Umständen mehrere Computerstarts vergehen, bis alle für einen Rechner benötigten Updates und Softwarepakete installiert sind.

Nach vierjährigem Einsatz der Lösung, wuchs die Anzahl der programmierten Skripte auf eine unübersichtliche und sehr wartungsintensive Menge an die sich zudem als sehr unflexibel erwies. Eine angestrebte Auslieferung von Daten und Programmen in Echtzeit war hierüber nicht möglich. Eine Weiterentwicklung der Inhouse-Lösung erschien nicht erfolgversprechend. Neue Verfahren sollten die oben beschriebenen Prozesse vereinfachen und zusammen-

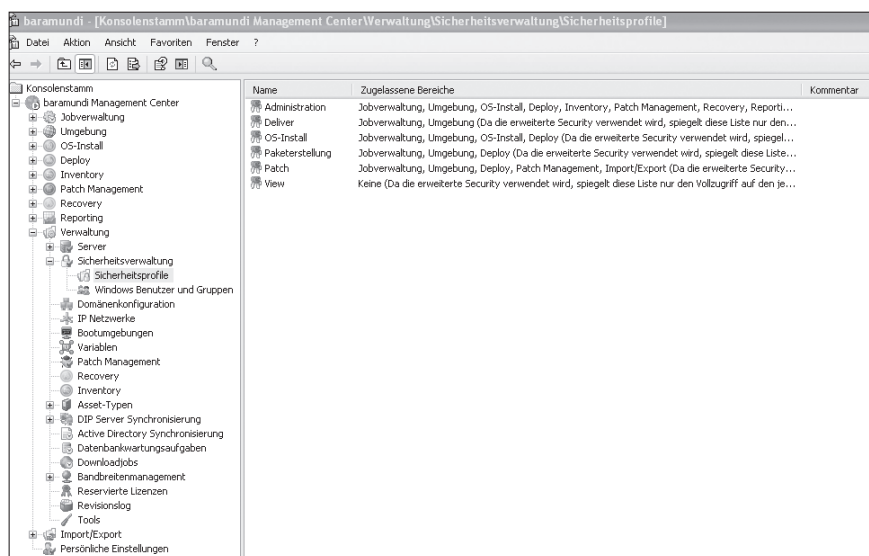


Abb. 2: Baramundi Oberfläche

für die professionelle Verwaltung eines Windows-Netzwerkes notwendig sind. Für den Einsatz der Suite werden ein Windows-Server sowie eine Datenbank benötigt. Die Datenbank beinhaltet neben den Inventarisierungsdaten die Daten der hinterlegten Jobs, Applikationen, Patchinformationen und die Konfigurationsdaten des Client-Management-Servers selbst. Aus Performance-Gründen sowie zur besseren Absicherung erfolgt der Datenbankbetrieb separat. Für das Testsystem in der Abteilung „DV in der Verwaltung“ wurden auf der Basis von Windows 2003 Server der „baramundi-Management-Server“ (bms) und ein Datenbankserver mit Microsoft SQL Server implementiert. Dabei kann es sich um physikalische Hardware, aber auch um virtuelle Maschinen handeln. Zu Testzwecken sind beide Varianten implementiert worden. Die Mindestanforderungen für die Hardware, ein aktueller Prozessor mit zwei Kernen, eine 100 GB Festplatte, 4 GB Arbeitsspeicher sowie eine 1 Gigabit Netzwerkkarte finden in den Testsystemen Verwendung. Die Schnittstelle zur Steuerung des Datenaustausches zwischen der Datenbank und dem „bms“ bildet der „Management Object Connector“ (MOC). Der „bms“ kommuniziert mit dem „MOC“, um Befehle und Daten für die Datenbank entgegenzunehmen. Als weitere Komponente wird ein „Distributed Installation Point“ (DIP) benötigt. Der „DIP“ stellt ein Depot zur Verteilung von Software dar. Für die Testinstallation in der Abteilung

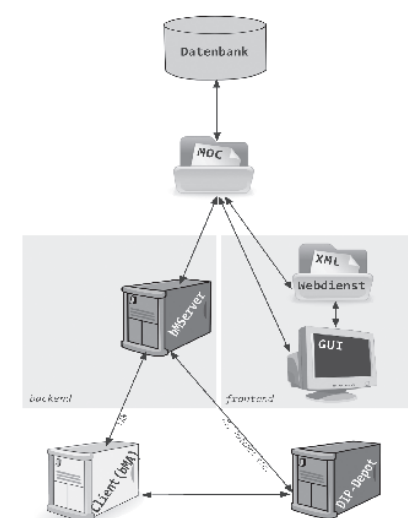


Abb. 3: Baramundi-Topologie

„DV in der Verwaltung“ wurde der „DIP“ über eine Windows-Freigabe realisiert. Des Weiteren steuert der „bms“ die Kommunikation mit den Clients. Der Client kommuniziert mit dem „bms“ über den „baramundi-Management-Agent“ (bma). Dieser Agent ist auf jedem Rechner installiert.

Nachfolgende Grafiken verdeutlicht den Aufbau bzw. die Kommunikation zwischen den Komponenten der „baramundi-Management-Suite“:

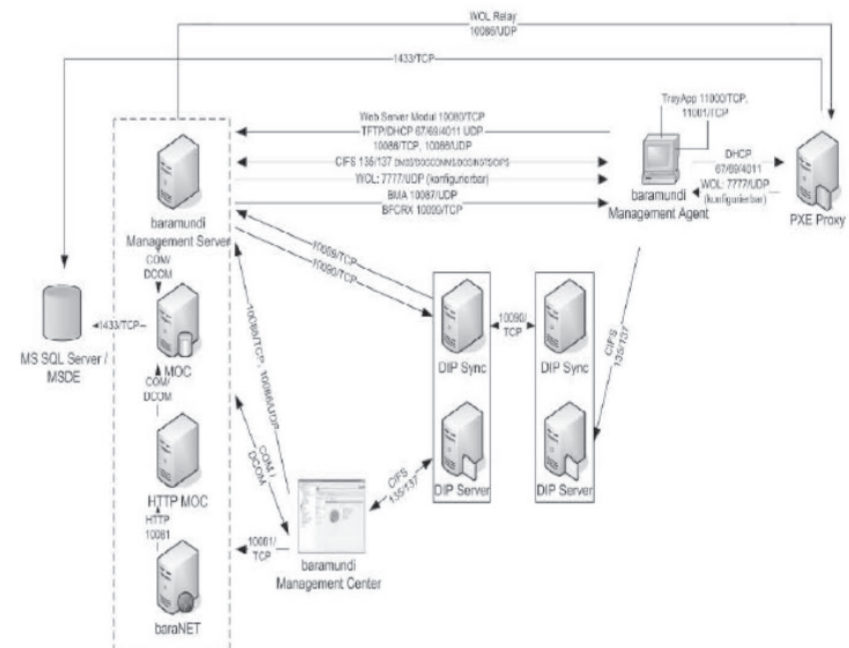


Abb. 4: Baramundi-Kommunikationsschema

Software-Management

Die Baramundi Module „OS-Install“ und „Deploy“ ermöglichen die vollautomatische Installation von Clients über das Netzwerk. Mit Hilfe der Module für das Softwaremanagement wird die Wartung des Betriebssystems und weiterer Applikationen auf einer Auswahl von unterschiedlichen Clients möglich. Die Module gewährleisten eine weitgehende Delegation von Funktionen und Rollen an administrative Nutzergruppen. Die Automatisierung der Installationsprozesse erhöht zum einen die Qualität der Softwareverteilung und zum anderen reduziert sie langfristig den Zeit- und Ressourcenaufwand.

Baramundi „OS-Install“ ermöglicht, die Betriebssystem-Installation vollautomatisch vorzunehmen. Clients können in

nur einem Schritt partitioniert, formatiert, nativ installiert und konfiguriert werden.

Ein stufenloser Übergang zum Modul „Deploy“ ermöglicht die automatisierte und zeitgenaue Verteilung der Software. Bestandteil des Moduls „Deploy“ ist die Paketerstellung von Softwareprodukten. Bei der Paketerstellung wird die Software so vorbereitet, dass ein automatischer Installations- und Deinstallationsprozess ohne Benutzereingriffe möglich

ist. In den Softwarepaketen werden spezifische Voreinstellungen und Konfigurationen für die Nutzer hinterlegt. So lassen sich zum Beispiel bei der Konfiguration eines Softwarepaketes Abhängigkeiten zu anderer Software definieren. Diese Abhängigkeiten stellen sicher, dass alle Voraussetzungen für die Installation einer Software gegeben sind. Beispiel hierfür ist die Installation einer aktuellen Java Runtime Environment (JRE) Version vor der Installation von OpenOffice.

Die Auslieferung der Softwarepakete wird über das Modul „Jobverwaltung“ gesteuert. Ein Job besteht zum Beispiel aus der Installation des Betriebssystems oder einer Software wie OpenOffice. Realisierbar ist auch die Bündelung mehrerer Softwarepakete zu einem Job. Des Weiteren ist eine zeitlich versetzte, sofortige oder zyklische Ausführung der Aufgaben

Tables	Directory	Directory Parent	DefaultDir
ActionText	INSTALLLOCATION	ProgramFilesFolder	.
AdminExecuteSequence	OpenOfficeorg3_Basis_program_python_core_261_lib_encodings	OpenOfficeorg3_Basis_program_python_core_261_lib	ENCODING~1 encodings
AdminUISequence	OpenOfficeorg3_program	OpenOfficeorg3	program
AdvExecuteSequence	OpenOfficeorg3_Basis_share_registry_schema_org_openoffice	OpenOfficeorg3_Basis_share_registry_schema_org	OPENVOP~5 openoffice
AppSearch	OpenOfficeorg3_Basis_share_registry_data_org_openoffice_Office	OpenOfficeorg3_Basis_share_registry_data_org_openoffice	Office
Binary	OpenOfficeorg3_Basis_share_Scripts_beanshell_Highlight	OpenOfficeorg3_Basis_share_Scripts_beanshell	HIGHLIGHT~1 Highlight
Checkbox	FontsFolder	OpenOfficeorg3_Basis_share_fonts	Fonts:trueuetype
Component	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	Types
Control	OpenOfficeorg3_Basis_program_classes	OpenOfficeorg3_Basis_program	classes
ControlCondition	OpenOfficeorg3_Basis_help_de	OpenOfficeorg3_Basis_help	de
ControlEvent	OpenOfficeorg3_URI_bin	OpenOfficeorg3_URI	bin
CreateFolder	OpenOfficeorg3_Basis_share_fingerprint	OpenOfficeorg3_Basis_share	FINGER~1 fingerprint
CustomAction	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbtdata_me...	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbtdata	menubar
Dialog	OpenOfficeorg3_Basis_share_registry_data_org_openoffice_Office_UI	OpenOfficeorg3_Basis_share_registry_data_org_openoffice_Office	UI
Directory	OpenOfficeorg3_Basis_share_gallery_hnlexpo	OpenOfficeorg3_Basis_share_gallery	hnlexpo
Error	OpenOfficeorg3_Basis_program_resource	OpenOfficeorg3_Basis_program	resource
EventMapping	OpenOfficeorg3_Basis_share_basic_Schedule	OpenOfficeorg3_Basis_share_basic	Schedule
Feature	OpenOfficeorg3_Basis_program_python_core_261_lib_email	OpenOfficeorg3_Basis_program_python_core_261_lib	email
FeatureComponents	OpenOfficeorg3_Basis_share_template_de_layout	OpenOfficeorg3_Basis_share_template_de	layout
File	OpenOfficeorg3_Basis_share_xslt_export_wordml	OpenOfficeorg3_Basis_share_xslt_export	wordml
Font	OpenOfficeorg3_Basis_share_registry_schema_org_openoffice_Office...	OpenOfficeorg3_Basis_share_registry_schema_org_openoffice_Office	UI
Icon	OpenOfficeorg3_Basis_program_python_core_261_lib_compiler	OpenOfficeorg3_Basis_program_python_core_261_lib	compiler
Infrafile	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	Filter
InstallExecuteSequence	OpenOfficeorg3_Basis_share_basic_Depot	OpenOfficeorg3_Basis_share_basic	Depot
InstallUISequence	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_Office...	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_Office	Logging
LaunchCondition	OpenOfficeorg3_URI_java	OpenOfficeorg3_URI	java
Listbox	OpenOfficeorg3_Basis_presets_database	OpenOfficeorg3_Basis_presets	database
Media	OpenOfficeorg3_Basis_share_config_softicedg_global_accelerator_es	OpenOfficeorg3_Basis_share_config_softicedg_global_accelerator	es
ModuleAdminExecuteSequence	OpenOfficeorg3_Basis_share_template_wizard_letter_zh_CN	OpenOfficeorg3_Basis_share_template_wizard_letter	zh-CN
ModuleAdvExecuteSequence	OpenOfficeorg3_Basis_share_registry_schema_org_openoffice_Office	OpenOfficeorg3_Basis_share_registry_schema_org_openoffice	Office
ModuleComponents	OpenOfficeorg3_Basis_share_basic_Template	OpenOfficeorg3_Basis_share_basic	Template
ModuleDependency	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	TYPEDE~4 TypeDetection
ModuleInstallExecuteSequence	OpenOfficeorg3_Basis_share_xslt_export_spreadsheetml	OpenOfficeorg3_Basis_share_xslt_export	SPREAD~1 spreadsheetml
ModuleSignature	OpenOfficeorg3_Basis_share_gallery_www_graf	OpenOfficeorg3_Basis_share_gallery	www-graf
MsAssembly	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbrelation_t...	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbrelation	toolbar
MsAssemblyName	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbbrowser_...	OpenOfficeorg3_Basis_share_config_softicedg_modules_dbbrowser_acc...	en-GB
MsDigitalCertificate	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_TypeDe...	Misc
MsDigitalSignature	OpenOfficeorg3_Basis_program_python_core_261_lib_test	OpenOfficeorg3_Basis_program_python_core_261_lib	test
MsFileHash	OpenOfficeorg3_Basis_share_registry_data_org_openoffice	OpenOfficeorg3_Basis_share_registry_data_org	OPENVOP~2 openoffice
MsFCBypass	OpenOfficeorg3_Basis_share_template_wizard_letter_it	OpenOfficeorg3_Basis_share_template_wizard_letter	it
Property	OpenOfficeorg3_Basis_share_dtd_officedocument_10	OpenOfficeorg3_Basis_share_dtd_officedocument	1_0
RadioButton	OpenOfficeorg3_Basis_share_template_wizard_letter_pt	OpenOfficeorg3_Basis_share_template_wizard_letter	pt
Reg4	OpenOfficeorg3_Basis_program	OpenOfficeorg3_Basis	program
RegLocator	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_Office_Jobs	OpenOfficeorg3_Basis_share_registry_modules_org_openoffice_Office	Jobs
Registry	OpenOfficeorg3_Basis_share_config_softicedg_modules_sweb_accele...	OpenOfficeorg3_Basis_share_config_softicedg_modules_sweb_accelerator	en-US
RemoveFile	OpenOfficeorg3_Basis_share_config_wizard_web_layouts_simple	OpenOfficeorg3_Basis_share_config_wizard_web_layouts	simple
RegFile	OpenOfficeorg3_Basis_share_template_de_wizard_report	OpenOfficeorg3_Basis_share_template_de_wizard	report
Shortcut	OpenOfficeorg3_Basis_share_Scripts_Javascript_ExportSheetsToHTML	OpenOfficeorg3_Basis_share_Scripts_Javascript	EXPORT~1~1 ExportSheetsToHTML
Signature	OpenOfficeorg3_Basis_share_config_wizard_web_buttons	OpenOfficeorg3_Basis_share_config_wizard_web	buttons
SizeMemGenComponents	OpenOfficeorg3_Basis_share_xslt_export_uof	OpenOfficeorg3_Basis_share_xslt_export	uof
TextStyle	OpenOfficeorg3_Basis_share_Scripts_Java_MemoryUsage	OpenOfficeorg3_Basis_share_Scripts_Java	MEMORY~2 MemoryUsage
UIText	OpenOfficeorg3_Basis_share_config_softicedg_modules_swreport_to...	OpenOfficeorg3_Basis_share_config_softicedg_modules_swreport	toolbar
Upgrade	OpenOfficeorg3_Basis_share_config_softicedg_modules_swreport_acce...	OpenOfficeorg3_Basis_share_config_softicedg_modules_swreport_acce...	de
_Validation	OpenOfficeorg3_Basis_share_xslt_import_wordml	OpenOfficeorg3_Basis_share_xslt_import	wordml

Abb. 5: Ausschnitt aus der angepassten MSI-Datei für OpenOffice

möglich. Beispiele für eine zyklische Jobausführung sind die Inventarisierung der Hard- und Software oder die Verteilung von Patches. Durch die Integration des Active Directory¹ in die „baramundi-Management-Suite“ können Aufgaben einzelnen Rechnern, Nutzern oder Gruppen zugeordnet werden, wobei es sich um Active Directory¹ oder um dynamische Gruppen, mittels SQL-Statements gebildet, handeln kann.

OpenOffice

Am Beispiel der freien Bürosoftware OpenOffice wird das Verfahren der Software-Verteilung über Baramundi vorge-

stellt. Da in der zentralen Universitätsverwaltung der Humboldt-Universität standardmäßig Microsoft Office 2003 installiert ist, erfolgt die Verteilung der Software OpenOffice ausschließlich auf Nutzeranfrage an spezielle Clients. Die Systemvoraussetzungen für den Einsatz dieser Software unter dem Betriebssystem Windows werden auf den Webseiten von OpenOffice.org wie folgt beschrieben:

- Windows 2000 ab SP 2 oder höhere Windows-Version
- mindestens 256 MB RAM, 512 MB empfohlen
- mindestens 650 MB freier Festplattenplatz
- 1024 x 768 Auflösung mit mindestens 256 Farben

Software verteilen

Vor Erstellung des Installationspaketes werden alle Installationsdateien auf den „DIP“ hinterlegt. Die Erstellung des Installations- bzw. Deinstallationspaketes im Modul „Deploy“ geschieht mit oder ohne Wizard-Unterstützung. Beim Installationspaket kann es sich um ein

Skript, das mit Hilfe des Baramundi-Automation-Studio² erstellt wurde, eine vorhandene MSI/MST-Datei³ oder ein Skript in einer beliebigen Skriptsprache handeln. Da OpenOffice für die Installation der Software MSI-Dateien verwendet, erfolgt die Erstellung des Installationspaketes für die zentrale Verwaltung der Humboldt-Universität mittels einer angepassten MSI-Datei.

Die Anpassungen erfolgen mit dem Orca-Datenbank-Editor von Microsoft. In dem verwendeten MSI-File werden folgende Einstellungen konfiguriert:

- Standardverzeichnisse für das Speichern und Laden von Dateien
- Zugriff auf das Java Runtime Environment
- Einstellungen für die Makro-Sicherheit

1 Active Directory (AD): Verzeichnisdienst von Microsoft Windows Server. Active Directory ermöglicht es, ein Netzwerk entsprechend der realen Struktur des Unternehmens oder seiner räumlichen Verteilung zu gliedern. Dazu verwaltet es verschiedene Objekte, wie beispielsweise Benutzer, Gruppen, Computer, Dienste, Server, Dateifreigaben und andere Geräte wie Drucker und Scanner und deren Eigenschaften. Mit Hilfe von Active Directory kann ein Administrator die Informationen der Objekte organisieren, bereitstellen und überwachen. Den Benutzern des Netzwerkes können Zugriffsbeschränkungen erteilt werden. So darf zum Beispiel nicht jeder Benutzer jede Datei sehen oder jeden Drucker verwenden.

2 Baramundi-Automation-Studio: Werkzeug zur Erstellung von Skripten für Administrationsaufgaben oder zur Oberflächenautomatisierung

3 MSI/MST-Datei: MSI-Format (Microsoft Software Installation) sowie MST für Transform-Dateien Windows Installer Datei (vormals Microsoft Installer) stellen eine Laufzeitumgebung für Installationsroutinen unter Microsoft-Windows-Betriebssystemen bereit. Sie besteht aus einem Windows-Systemdienst, welcher Paketdateien interpretieren, entpacken und ausführen kann.

- Einstellungen für Standardvorlagen und für HU-spezifische Vorlagen wie die Arbeitszeiterfassung
- Aktivierung des Browser-PlugIns für Office-Dokumente
- Festlegung des Standard-Formates für das Speichern von Dokumenten

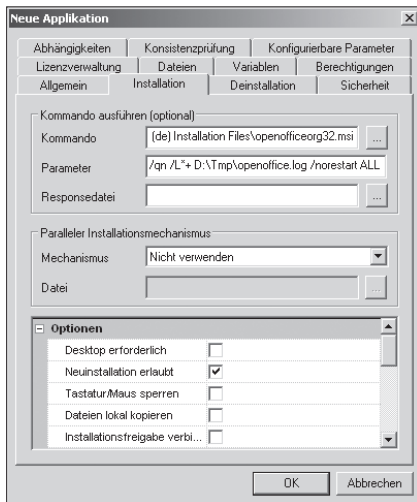


Abb. 6: Screenshot eines Deploy-Skripts

Die Auslieferung des Softwarepaketes für OpenOffice geschieht mittels eines „Jobs“ im Modul „Jobverwaltung“ der „baramundi-Management-Suite“. Mit Hilfe des „Jobs“ wird dem Client die Applikation zugeordnet.

Abbildung 7 zeigt den in der Jobverwaltung definierten Job für das Jobpaket OpenOffice unter Verwendung eines MSI-Files. Über Returncodes wird während der Installation geprüft, ob die Software fehlerfrei installiert wurde.

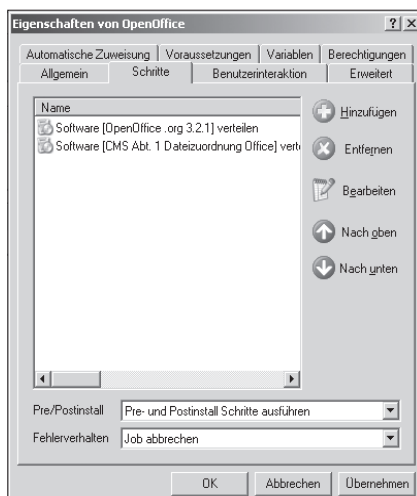


Abb. 7: Job-Skript

„Jobs“ können Installations- und Deinstallations-Routinen miteinander verknüpfen. Im Beispiel von OpenOffice wird mit Hilfe des Befehls „assoc“ von Microsoft eine Erweiterung der Office-Dateiendungen vorgenommen. Hierdurch lassen sich verschiedene Installationsvarianten von Microsoft-Office und OpenOffice auf den Rechnern erzeugen. Einerseits können beide Office-Applikationen auf einem Client installiert, andererseits die Installation von OpenOffice mit der Deinstallation von Microsoft Office verknüpft werden. Je nach Installationsvariante können alle Office-Dateien mit einer Installation geöffnet werden. Dies ist durch einen entsprechenden Eintrag in der MSI-Datei von OpenOffice möglich.

Auszug aus der Client-Log-Datei zur Verteilung von OpenOffice:

```
19.01.2011 14:16:19 Die Verbindung zum Server [HUXXX] wurde erfolgreich aufgebaut
19.01.2011 14:16:23 Job [OpenOffice] zur Ausführung vom Server erhalten
19.01.2011 14:16:23 Der Job [OpenOffice] wird verarbeitet
19.01.2011 14:16:23 Der Jobschritt [Applikation [OpenOffice] installieren] wird gestartet
19.01.2011 14:16:23 Die Installation von [OpenOffice] wird gestartet...
19.01.2011 14:16:23 Der Sicherheitskontext des baramundi Management Suite Installations-
benutzers [test@huberlin.de] wird verwendet.
19.01.2011 14:16:23 Der DIP wird aus den verfügbaren DIPs [\\huxxx\xxxdv\dip_xxx] ermittelt.
19.01.2011 14:16:23 Der DIP [\\huxxx\xxxdv\dip_xxx] wird verwendet
19.01.2011 14:16:23 Das baramundi Deploy Skript [\\huxxx.xxx.hu-berlin.de\xxdv\DIP_xxx\apl\
OpenOffice\OpenOffice.msi] wird ausgeführt
19.01.2011 14:16:23 Eingebettetes Skript wird zur Ausführung auf Platte gespeichert
19.01.2011 14:20:39 Die Konsistenzprüfung war erfolgreich.
19.01.2011 14:20:39 Der Wechsel des Sicherheitskontextes ist beendet
19.01.2011 14:20:39 Die Installation von [OpenOffice] war erfolgreich
19.01.2011 14:20:39 Das Job Step Packet [OpenOffice] wird gelöscht
19.01.2011 14:20:40 Die Ausführung von Job [OpenOffice] wurde beendet
```

Schlussbemerkung

Die geplante Überführung des Baramundi-Testsystems in den Produktivbetrieb erhöht die Qualität der Betreuung erheblich. Software und Updates werden schnell und präzise in Echtzeit an einzelne Nutzer, einzelne Clients, Clientgruppen oder alle Rechner ausgeliefert. Die Installation der Betriebssysteme in einem Schritt verkürzt die Installationszeit erheblich.

Perspektivisch erfolgt mittels der „Baramundi-Management-Suite“ eine Konzentration aller wichtigen Systembe-

treuungsaufgaben wie dem Patch-Management, der Installation des Betriebssystems und der Software-Verteilung in einem System. Hierdurch werden drei Systeme zur Verwaltung des Windows-Netzwerkes durch ein System ersetzt und so die vorhandenen Ressourcen zur Betreuung und Wartung ökonomischer eingesetzt.

Literatur

- [1] TOBIAS BEWRNDT: *Handbuch Baramundi Management Suite 8.0*. Baramundi Software AG, 2009, Seite 18ff, 134ff.
- [2] [HTTP://wpkg.org/OpenOffice.org_3.x](http://wpkg.org/OpenOffice.org_3.x)
- [3] [HTTP://user.services.openoffice.org/en/forum/viewtopic.php?f=74&t=28765&start=0](http://user.services.openoffice.org/en/forum/viewtopic.php?f=74&t=28765&start=0)
- [4] [HTTP://support.microsoft.com/kb/323526/en-us](http://support.microsoft.com/kb/323526/en-us)